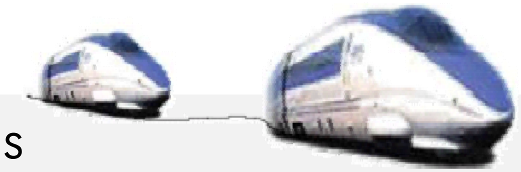
# Stochastic, Hybrid and Real-Time Systems: From Foundations To Applications with Modest



Holger Hermanns, Arnd Hartmanns

Saarland University, Germany

based on joint work with

Jonathan Bogdoll, Henrik Bohnenkamp, Pedro R. D'Argenio,
Alexandre David, Ernst Moritz Hahn, and Joost-Pieter Katoen

# Systems, Models and Requirements

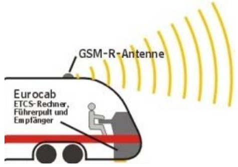
*All models are wrong, but some models are useful.*

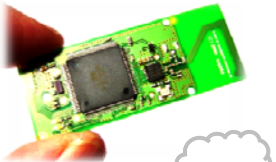
(George E. P. Box)

Model ⟷ model checking ⟷ Requirements

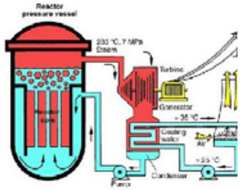
✓ ✘

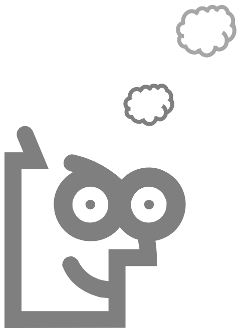
this is what we want

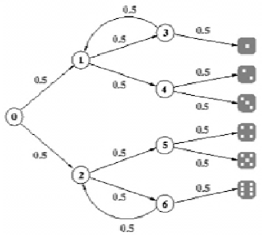


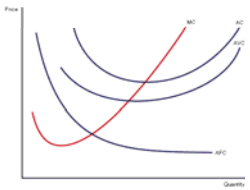


System under study / implementation









correctness safety performance costs

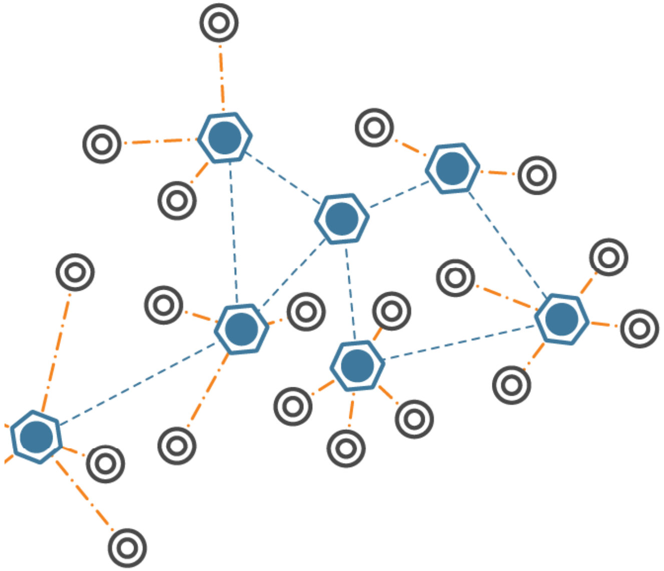
(slide inspired by Jan Tretmans, Embedded Systems Institute, Eindhoven)

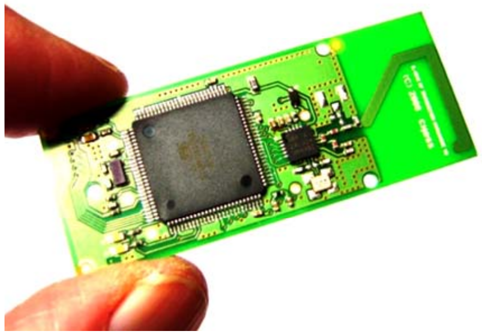
# Systems, Models and Requirements

*All models are wrong, but some models are useful.*

(George E. P. Box)

What are useful models?





*Wireless Sensor Networks:*

concurrency

message loss

transmission delays

randomised algorithms

limited battery power

# Systems, Models and Requirements

*All models are wrong, but some models are useful.*

(George E. P. Box)

What are useful models?

ETCS Level 3:

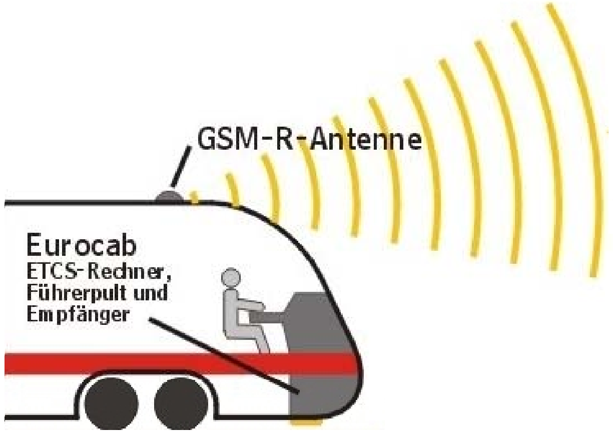
transmission delays

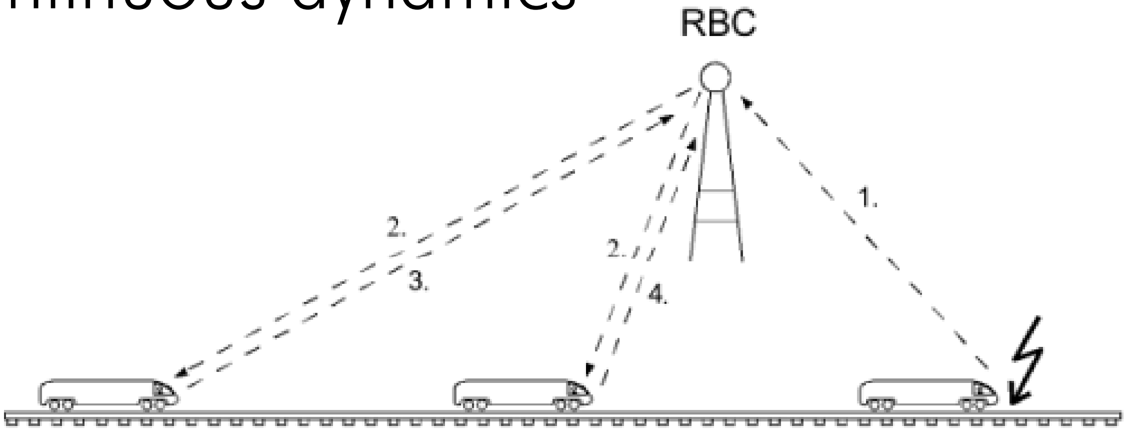
concurrency

message loss

measurement errors

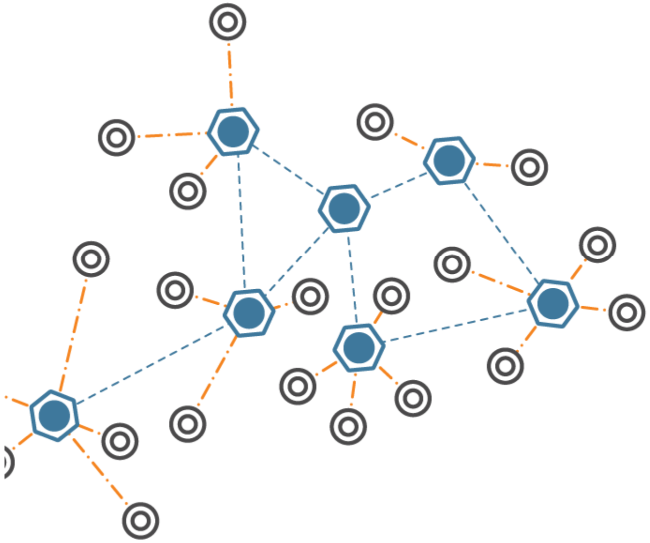
continuous dynamics





# Systems, Models and Requirements

*Quantitative models are useful.*



1 % probability of message loss

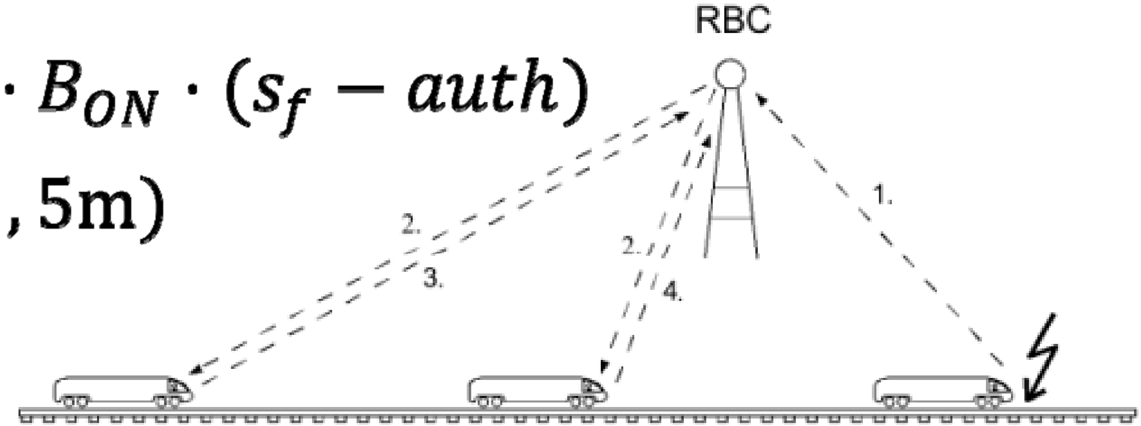
20 mW needed in send mode

Expected time for transmission $\leq$ 8 s ?

Fraction of time in send mode $\leq$ 0.2 ?

$$\dot{v} = a \wedge v \cdot v_{max} \leq 2 \cdot B_{ON} \cdot (s_f - auth)$$

$$pos_{seen} = \mathcal{N}(pos_{real}, 5\mathrm{m})$$



Prob(crash within 15 years) $\leq 10^{-5}$ ?

# Systems, Models and Requirements

*Quantitative models are useful.*

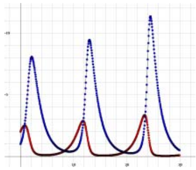
Quantities in models

time

probabilities

costs

continuous dynamics



$\dot{v} = a \wedge v \cdot v_1$

Quantities in requirements/properties

Quantified safety Prob(crash within 15 years) $\leq 10^{-5}$ ?

Performance Expected time for transmission $\leq$ 8 s ?

Dependability, Performability, Survivability, …

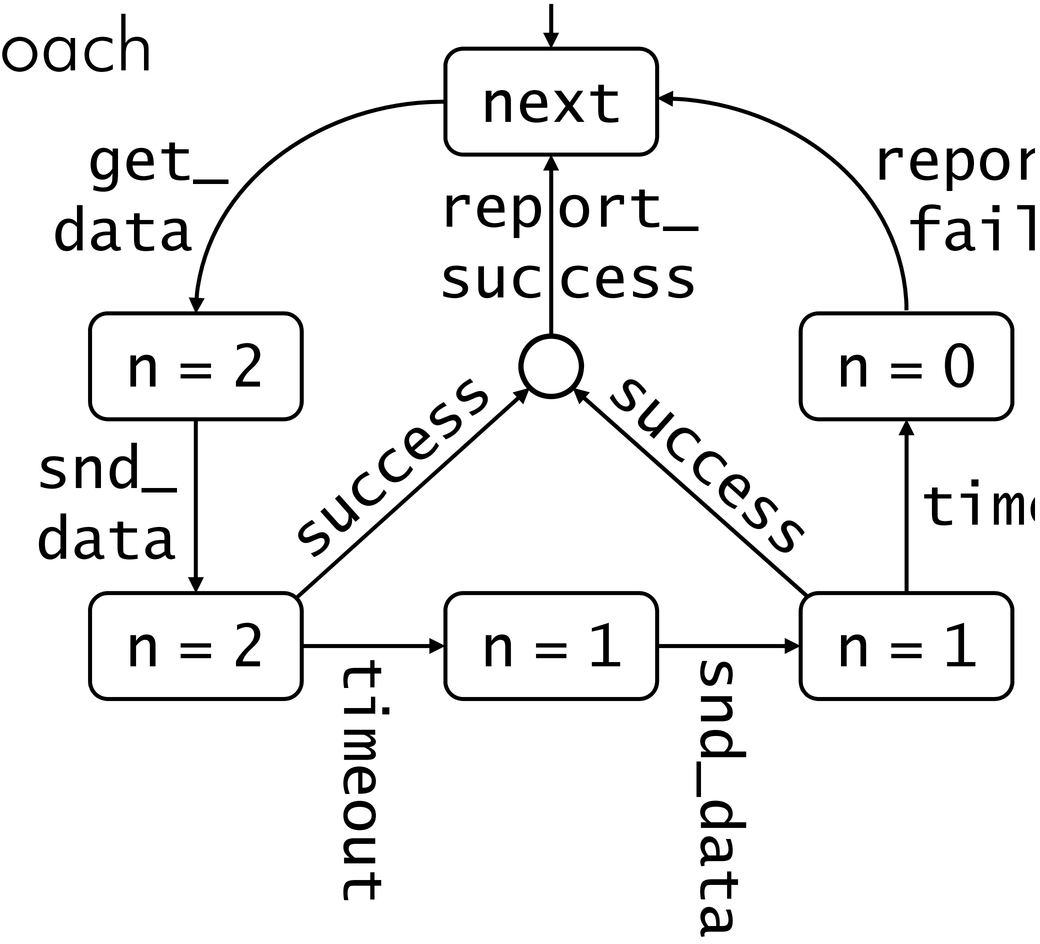
+ qualitative requirements in a quantitative setting

# Modelling and Verification

## The automata-based approach

**reactive system**

```
while(true)
  next:
  get_data(buf);
  n = 2;
  while(n > 0)
    e = snd_data(buf);
    if(e == SUCCESS)
      report_success();
      goto next;
    if(e == TIMEOUT)
      n = n - 1;
  report_failure();
```



# Modelling and Verification

The automata-based approach

Properties of interest

– Absence of deadlocks

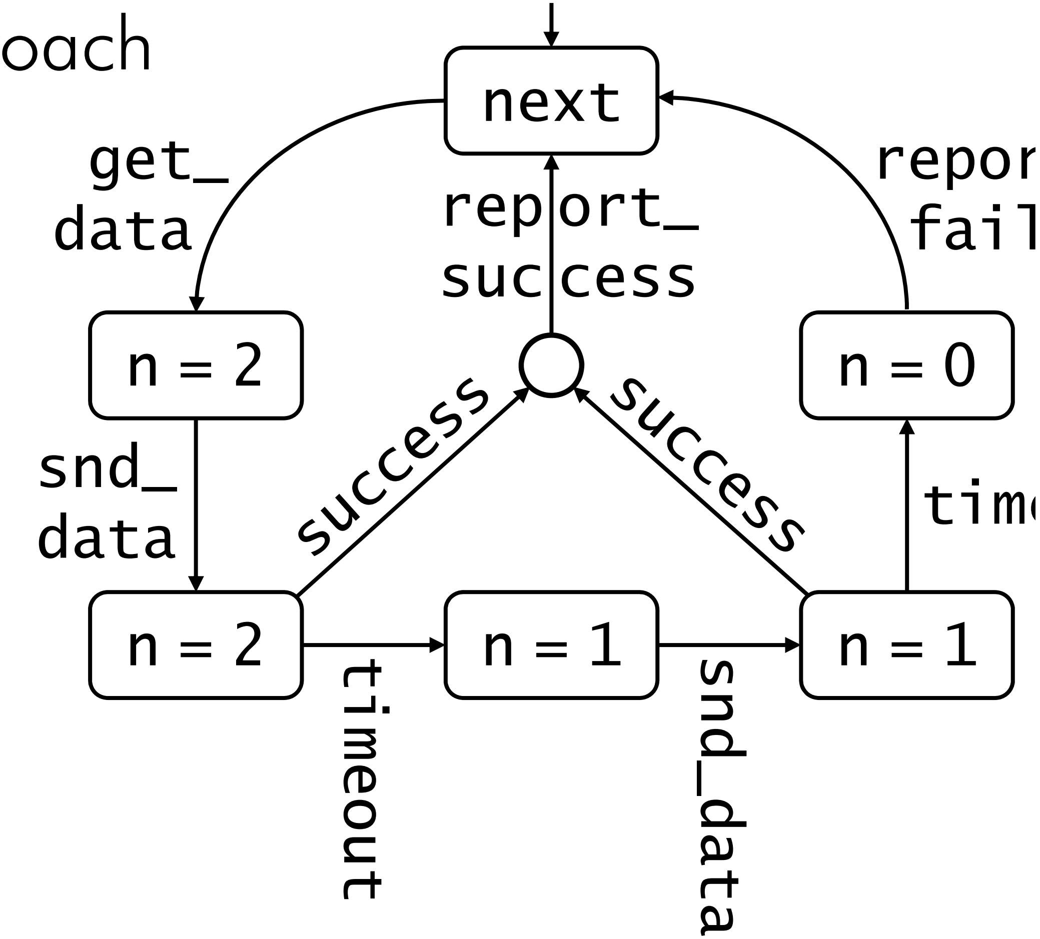
– Safety

– Liveness

– LTL or CTL formulas

e.g. $\forall\Box\exists\Diamond$ *success*

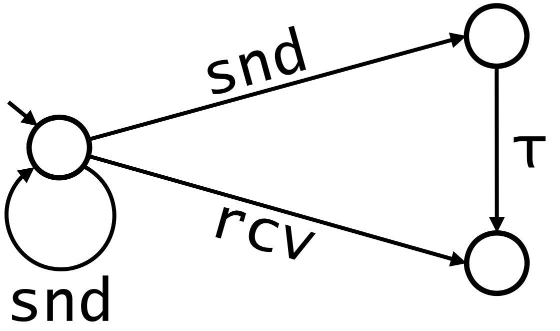
**Boolean requirements**



# Quantitative Models

A quantitative automata family

Labelled Transition Systems



LTS

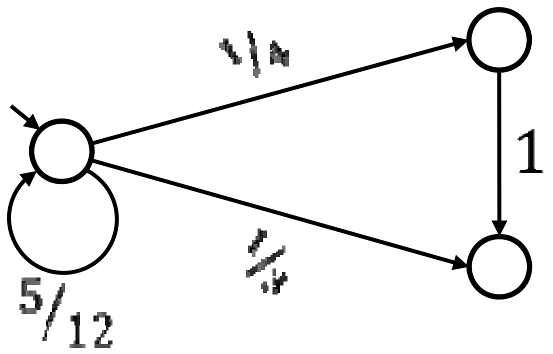
*nondeterminism*

# Quantitative Models

A quantitative automata family

Labelled Transition Systems

Discrete-Time Markov Chains



LTS

*nondeterminism*

DTMC

*discrete probabilities*

# Quantitative Models

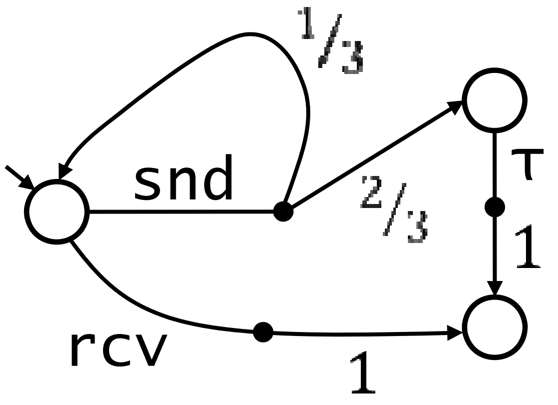
A quantitative automata family

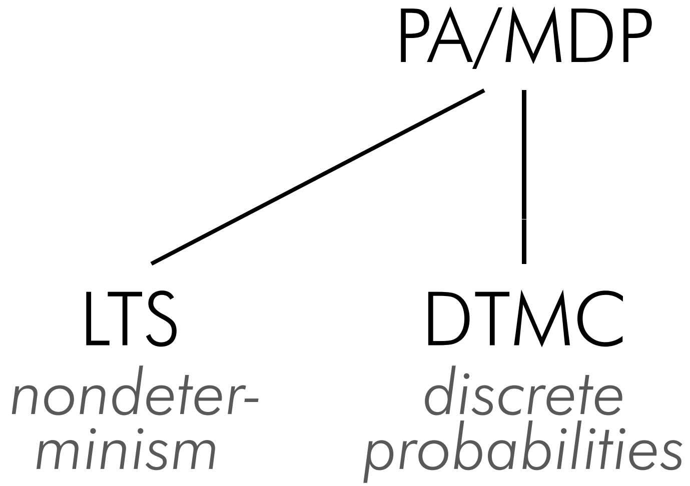
Labelled Transition Systems

Discrete-Time Markov Chains

Markov Decision Processes

Probabilistic Automata





PA/MDP

LTS
*nondeterminism*

DTMC
*discrete probabilities*

# Quantitative Models

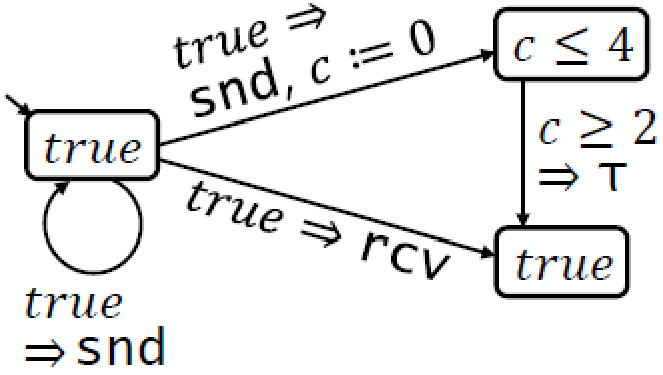
A quantitative automata family

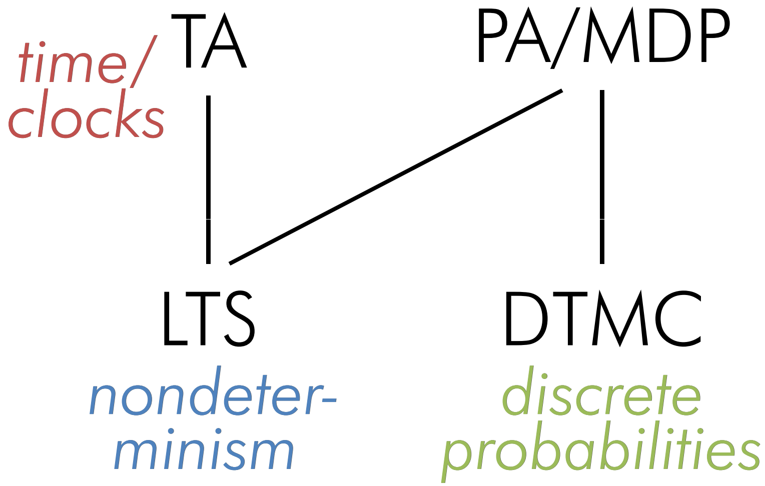
Labelled Transition Systems

Discrete-Time Markov Chains

Markov Decision Processes

Probabilistic Timed Automata





# Quantitative Models

A quantitative automata family

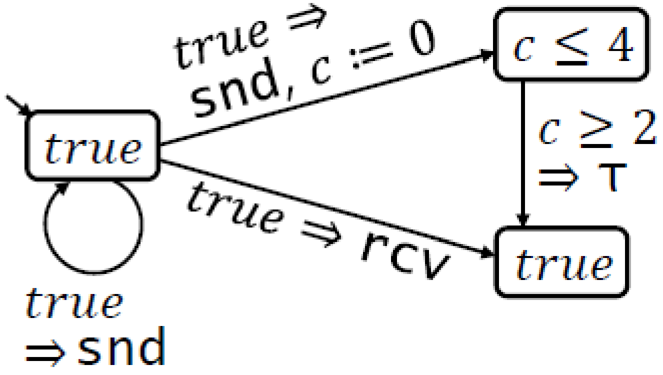
Labelled Transition Systems

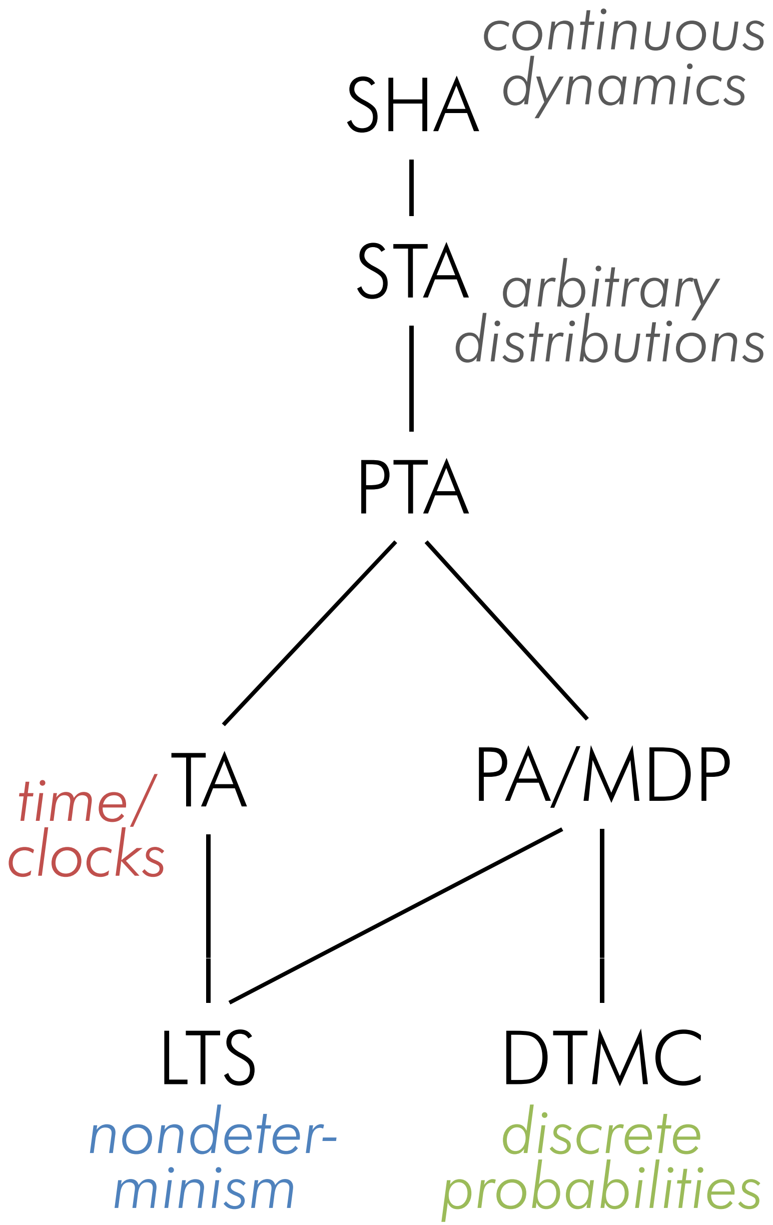
Discrete-Time Markov Chains

Markov Decision Processes

Probabilistic Timed Automata

Stochastic Timed /
Hybrid Automata




SHA

continuous dynamics

STA

arbitrary distributions

PTA

TA

time/ clocks

PA/MDP

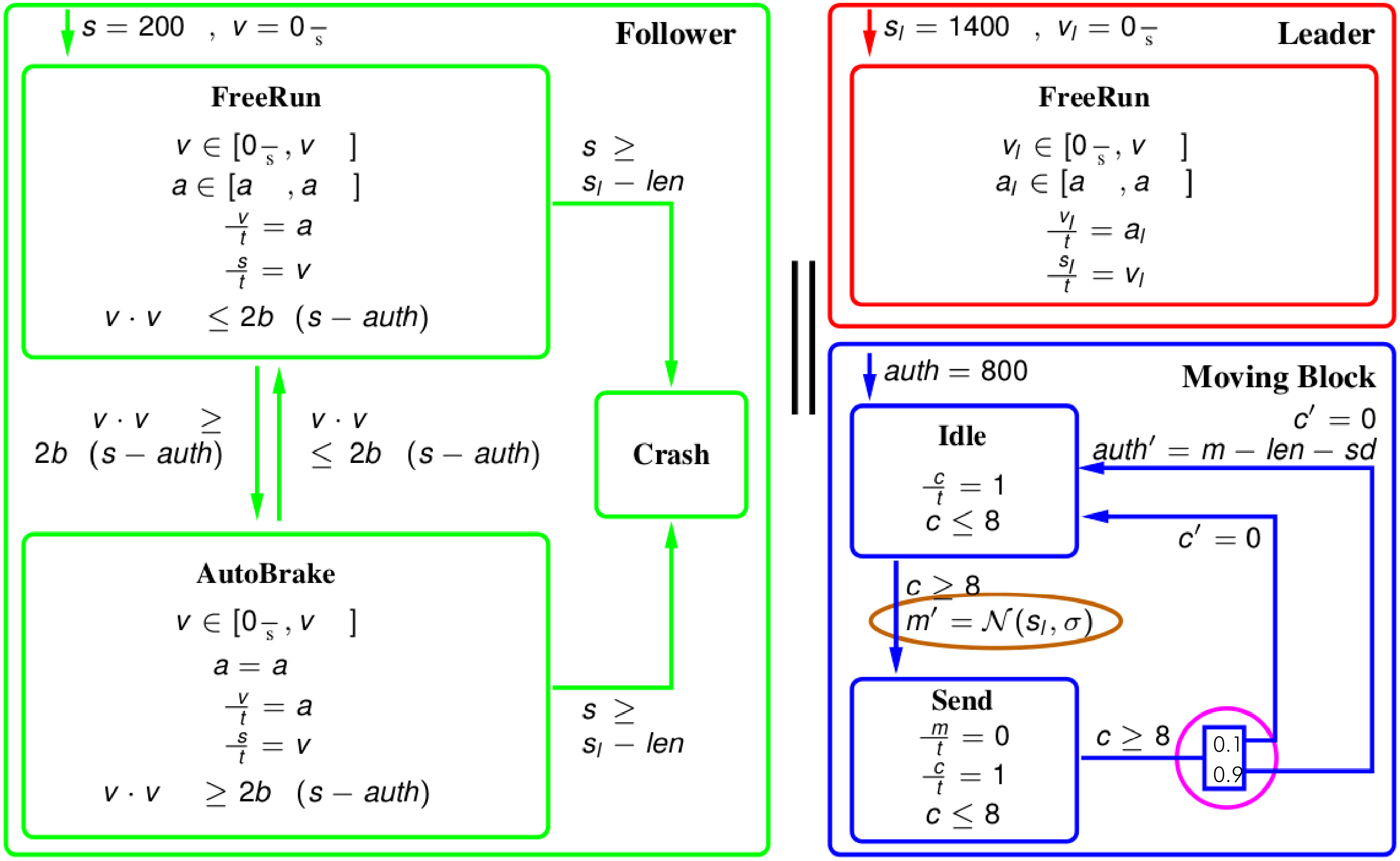
LTS

nondeterminism

DTMC

discrete probabilities

# A Stochastic Hybrid Automaton (Network)



Follower

$s = 200\ \ ,\ v = 0\frac{}{\mathrm{s}}$

FreeRun

$v \in [0\frac{}{\mathrm{s}}, v\ \ ]$

$a \in [a\ \ , a\ \ ]$

$\frac{v}{t} = a$

$\frac{s}{t} = v$

$v \cdot v\ \ \leq 2b\ \ (s - auth)$

$s \geq s_l - len$

$v \cdot v\ \ \geq 2b\ \ (s - auth)$

$v \cdot v\ \ \leq 2b\ \ (s - auth)$

Crash

AutoBrake

$v \in [0\frac{}{\mathrm{s}}, v\ \ ]$

$a = a$

$\frac{v}{t} = a$

$\frac{s}{t} = v$

$v \cdot v\ \ \geq 2b\ \ (s - auth)$

$s \geq s_l - len$

$\|$

Leader

$s_l = 1400\ \ ,\ v_l = 0\frac{}{\mathrm{s}}$

FreeRun

$v_l \in [0\frac{}{\mathrm{s}}, v\ \ ]$

$a_l \in [a\ \ , a\ \ ]$

$\frac{v_l}{t} = a_l$

$\frac{s_l}{t} = v_l$

Moving Block

$auth = 800$

Idle

$\frac{c}{t} = 1$

$c \leq 8$

$c' = 0$

$auth' = m - len - sd$

$c' = 0$

$c \geq 8$

$m' = \mathcal{N}(s_l, \sigma)$

Send

$\frac{m}{t} = 0$

$\frac{c}{t} = 1$

$c \leq 8$

$c \geq 8$

0.1

0.9

$v\ \ \ = 83.4\frac{}{\mathrm{s}},\ len = 200\ \ ,\ sd = 400\ \ ,\ a\ \ \ = -1.4\frac{}{\mathrm{s}^2},\ a\ \ \ = 0.7\frac{}{\mathrm{s}^2},\ b\ \ \ = -0.7\frac{}{\mathrm{s}^2},\ b\ \ \ = -0.3\frac{}{\mathrm{s}^2}$

# Quantitative Models

A quantitative automata family

Nondeterminism

– structural or temporal

Probabilistic choices

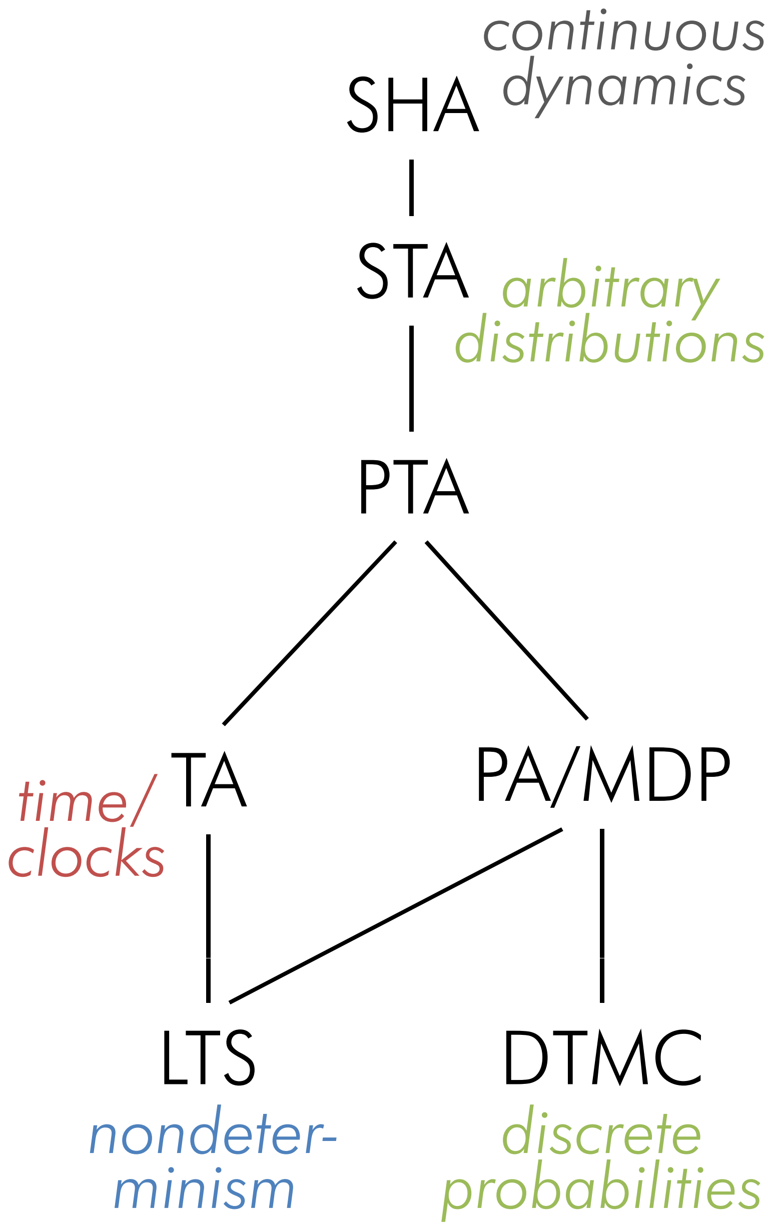
– discrete or continuous

– over next state or delay

Time

– discrete or continuous

– nondeterministic
 or random delays



# Quantitative Models

Automata modelling formalisms
and model checking tools

Modest
– The Modest Toolset

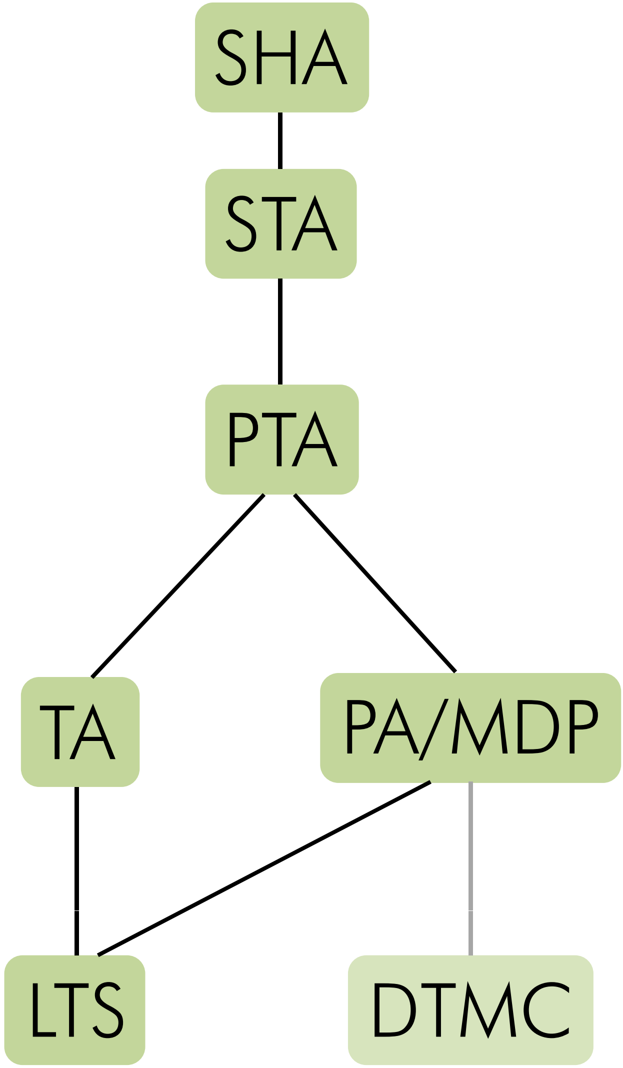
Guarded commands
– PRISM, PASS, …

"assembly language"

graphical

UPPAAL TA – UPPAAL

Promela etc – SPIN etc



# Models for Simulation

**Modest**: *A Modelling and Description Language for Stochastic Timed Systems*

Language features:

bool, int, arrays

Variables and assignments

Clocks

Processes and recursion

Rewards/costs

Exception handling

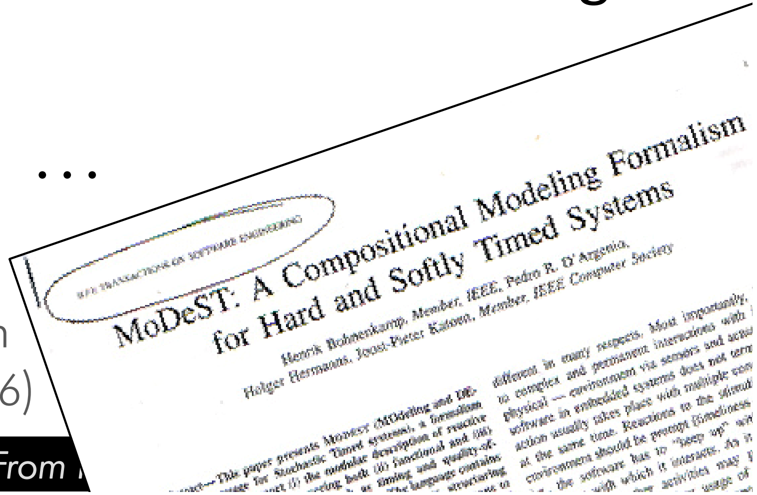
Deadlines & invariants

Probabilistic branching

Random variable sampling

…

Bohnenkamp, D’Argenio, Hermanns, Katoen:
MoDeST: A Compositional Modeling Formalism
for Hard and Softly Timed Systems (IEEE TSE 2006)



IEEE TRANSACTIONS ON SOFTWARE ENGINEERING

MoDeST: A Compositional Modeling Formalism for Hard and Softly Timed Systems

Henrik Bohnenkamp, Member, IEEE, Pedro R. D'Argenio, Holger Hermanns, Joost-Pieter Katoen, Member, IEEE Computer Society

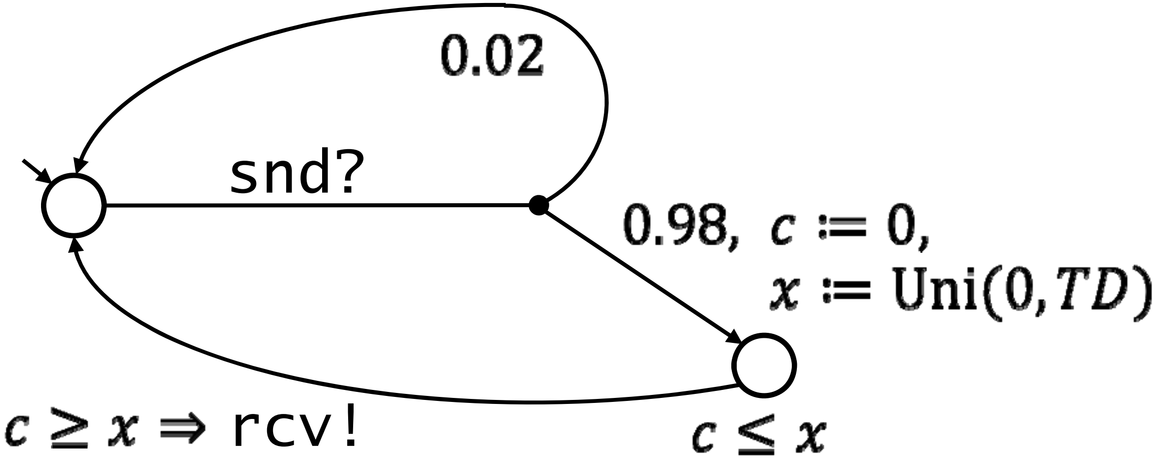
—This paper presents MODEST (MOdeling and DE-
guage for Stochastic Timed systems), a formalism
(i) the modular description of reactive
ering both (ii) functional and (iii)
as timing and quality-of-
The language contains
structuring

different in many respects. Most importantly,
to complex and permanent interactions with
physical — environment via sensors and actu
software in embedded systems does not term
action usually takes place with multiple con
at the same time. Reactions to the stimul
environment should be prompt (timeliness
the software has to "keep up" wi
with which it interacts. As it
activities may
usage of

# Example: Lossy channel with transmission delay

```
process Channel() {
  clock c;
  snd? palt {
    : 2: {==} // msg lost
    :98: {= c = 0, x = Uni(0, TD) =};
         invariant(c <= x) when(c >= x) rcv!
  };
  Channel()
}
```

**Stochastic Timed Automata Semantics**



# Modest – the language

high-level language

focus on readability, expressivity and conciseness

```
process Sender() {

   bool bit;
   int(0..MAX) rc;

   new_file {= i = 0, rc = 0 =};
   try {
      do {
      :: when(i < N) {= i = i + 1 =};
         do {
         :: put_k {= ff = (i == 1), lf = (i == N), ab = bit =}
            alt {
            :: get_l {= bit = !bit, rc = 0 =};
               break
            :: when(rc == MAX && i < N)
               s_nok {= rc = 0 =};
               throw(error)

            ...
```

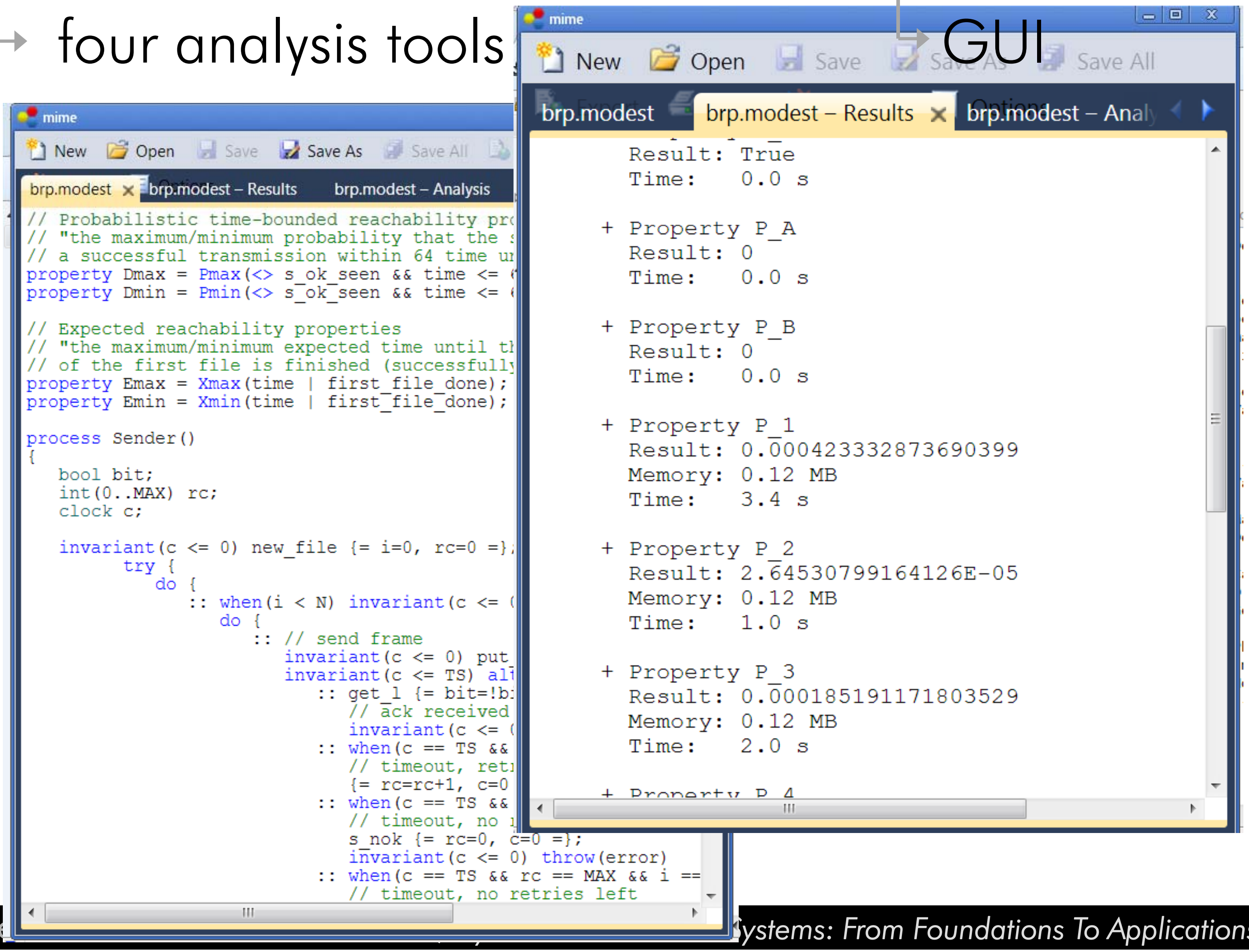
# The Modest Toolset

semantics

mctau – mcpta – prohver – modes – mime – mosta

four analysis tools

GUI

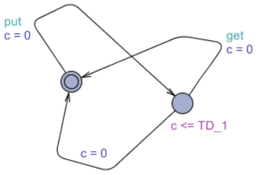


```
// Probabilistic time-bounded reachability pro
// "the maximum/minimum probability that the s
// a successful transmission within 64 time un
property Dmax = Pmax(<> s_ok_seen && time <= (
property Dmin = Pmin(<> s_ok_seen && time <= (

// Expected reachability properties
// "the maximum/minimum expected time until th
// of the first file is finished (successfully
property Emax = Xmax(time | first_file_done);
property Emin = Xmin(time | first_file_done);

process Sender()
{
   bool bit;
   int(0..MAX) rc;
   clock c;

   invariant(c <= 0) new_file {= i=0, rc=0 =};
         try {
            do {
               :: when(i < N) invariant(c <= (
                  do {
                     :: // send frame
                        invariant(c <= 0) put
                        invariant(c <= TS) alt
                           :: get_l {= bit=!bi
                              // ack received
                              invariant(c <= (
                           :: when(c == TS &&
                              // timeout, retr
                              {= rc=rc+1, c=0
                           :: when(c == TS &&
                              // timeout, no r
                              s_nok {= rc=0, c=0 =};
                              invariant(c <= 0) throw(error)
                           :: when(c == TS && rc == MAX && i ==
                              // timeout, no retries left
```

```
    Result: True
    Time:   0.0 s

  + Property P_A
    Result: 0
    Time:   0.0 s

  + Property P_B
    Result: 0
    Time:   0.0 s

  + Property P_1
    Result: 0.000423332873690399
    Memory: 0.12 MB
    Time:   3.4 s

  + Property P_2
    Result: 2.64530799164126E-05
    Memory: 0.12 MB
    Time:   1.0 s

  + Property P_3
    Result: 0.000185191171803529
    Memory: 0.12 MB
    Time:   2.0 s

  + Property P_4
```

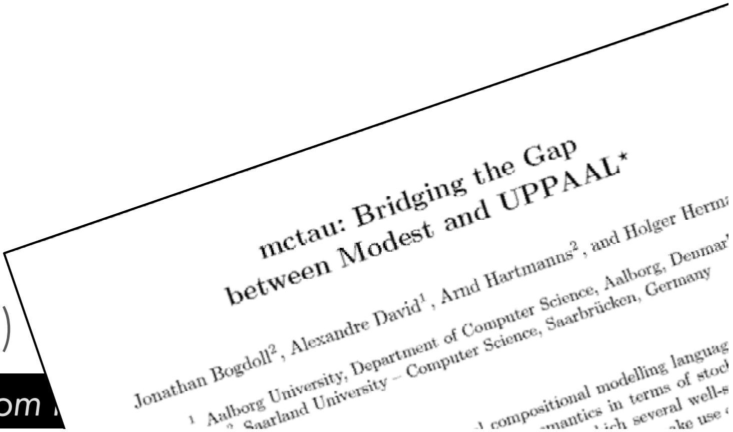
# The Modest Toolset

mctau – mcpta – prohver – modes – mime – mosta



mctau Model-checking for TA using UPPAAL
Export from Modest to UPPAAL with layout
Overapproximation of probabilistic choices

Bogdoll, David, H., H.: mctau: Bridging
the Gap between Modest and UPPAAL (SPIN 2012)



**mctau: Bridging the Gap
between Modest and UPPAAL***

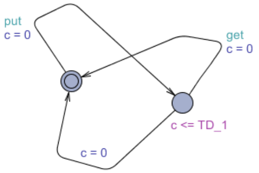
Jonathan Bogdoll[2], Alexandre David[1], Arnd Hartmanns[2], and Holger Herma

[1] Aalborg University, Department of Computer Science, Aalborg, Denmar
[2] Saarland University – Computer Science, Saarbrücken, Germany

compositional modelling languag
mantics in terms of stoc
ich several well-s
ke use

# The Modest Toolset

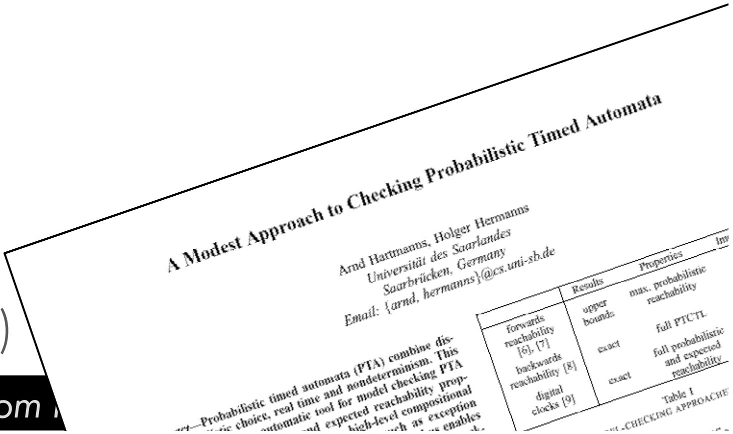
mctau – mcpta – prohver – modes – mime – mosta



**mctau** Model-checking for TA using UPPAAL
Export from Modest to UPPAAL with layout
Overapproximation of probabilistic choices

**mcpta** Model-checking for PTA using PRISM
Export from Modest to Guarded Commands

H., H.: A Modest Approach to
Checking Probabilistic Timed Automata (QEST 2009)



A Modest Approach to Checking Probabilistic Timed Automata

Arnd Hartmanns, Holger Hermanns
*Universität des Saarlandes*
*Saarbrücken, Germany*
*Email:* {*arnd, hermanns*}*@cs.uni-sb.de*

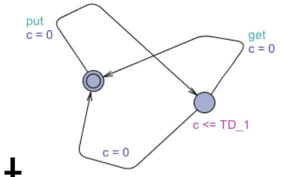
| | Results | Properties | In |
|---|---|---|---|
| forwards reachability [6], [7] | upper bounds | max. probabilistic reachability | |
| backwards reachability [8] | exact | full PTCTL | |
| digital clocks [9] | exact | full probabilistic and expected reachability | |

Table I
…EL-CHECKING APPROACHE…

…act—Probabilistic timed automata (PTA) combine dis-
…istic choice, real time and nondeterminism. This
…automatic tool for model checking PTA
…and expected reachability prop-
…high-level compositional
…uch as exception
…enables

# The Modest Toolset

mctau – mcpta – prohver – modes – mime – mosta

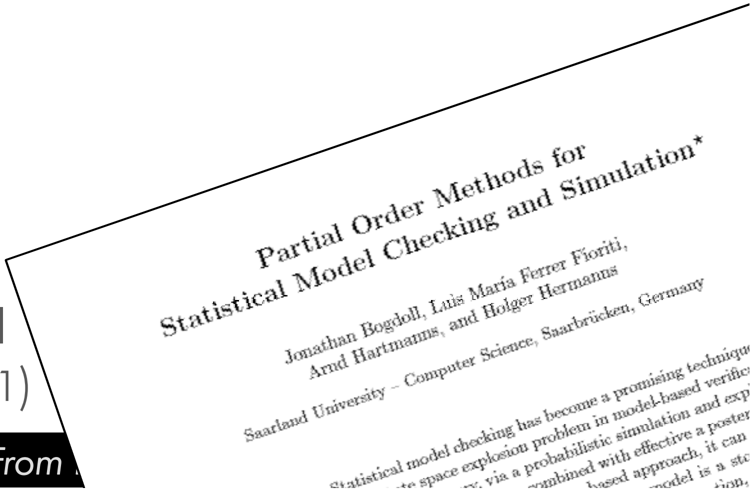


mctau Model-checking for TA using UPPAAL
Export from Modest to UPPAAL with layout
Overapproximation of probabilistic choices

mcpta Model-checking for PTA using PRISM
Export from Modest to Guarded Commands

modes Simulation & Statistical Model Checking for STA
with spurious nondeterminism

Bogdoll, Ferrer Fioriti, H., H.:
Partial Order Methods for Statistical Model
Checking and Simulation (FMOODS/FORTE 2011)



Partial Order Methods for
Statistical Model Checking and Simulation*

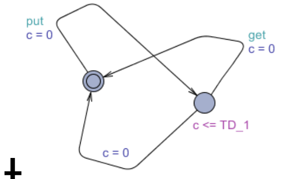
Jonathan Bogdoll, Luis María Ferrer Fioriti,
Arnd Hartmanns, and Holger Hermanns

Saarland University – Computer Science, Saarbrücken, Germany

Statistical model checking has become a promising techniqu
te space explosion problem in model-based verific
y, via a probabilistic simulation and exp
combined with effective a poster
based approach, it can
model is a sto
tion,

# The Modest Toolset

mctau – mcpta – prohver – modes – mime – mosta



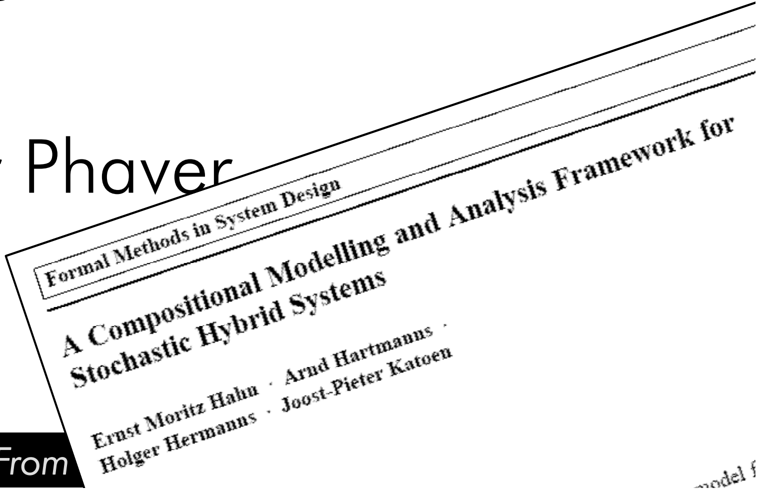
mctau Model-checking for TA using UPPAAL
Export from Modest to UPPAAL with layout
Overapproximation of probabilistic choices

mcpta Model-checking for PTA using PRISM
Export from Modest to Guarded Commands

modes Simulation & Statistical Model Checking for STA
with spurious nondeterminism

prohver Safety Verification for SHA
Using (modified) HA Solver Phaver

Hahn, H., H., Katoen:
A Compositional Modelling and Analysis Framework
For Stochastic Hybrid Systems (FMSD 13)



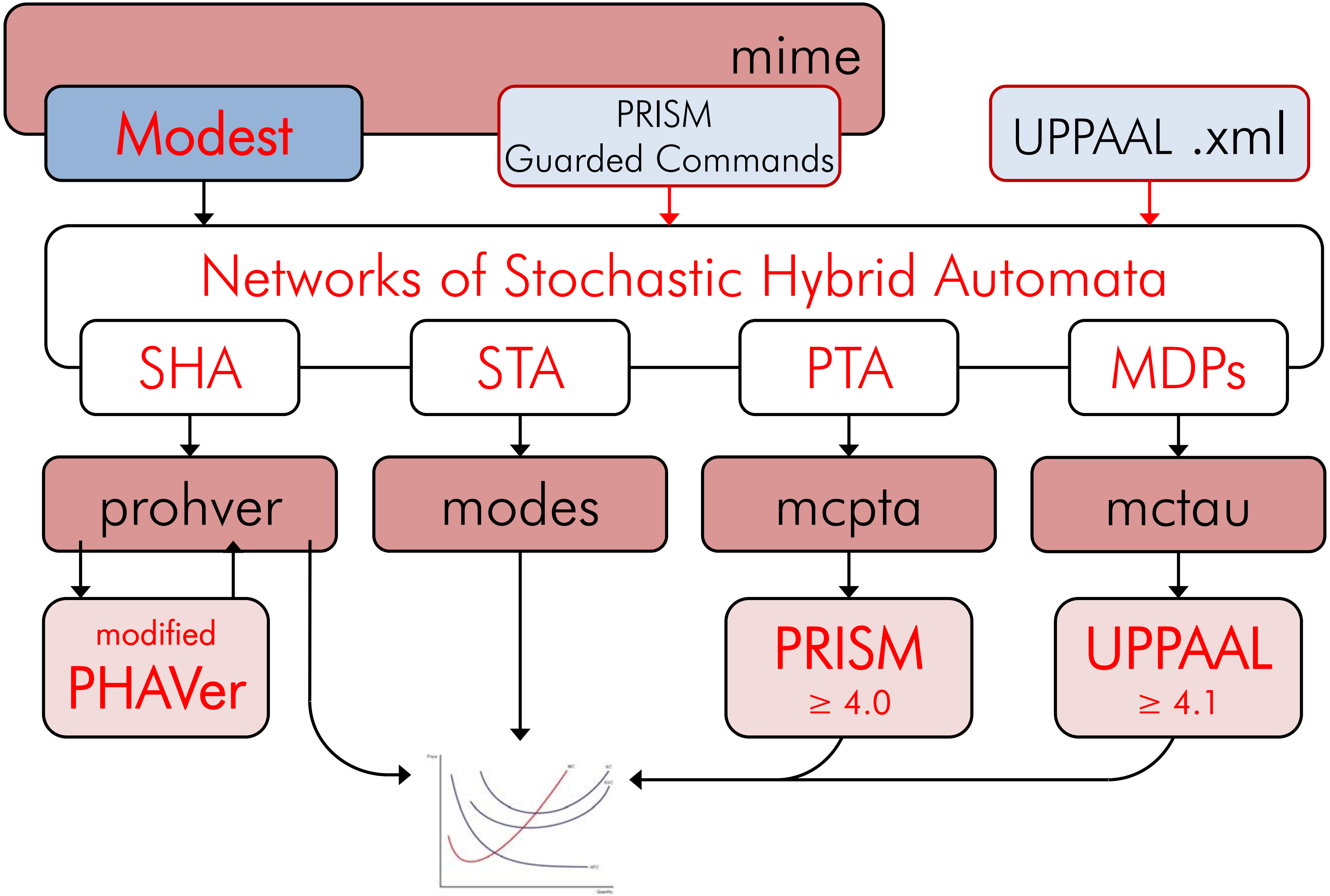
Formal Methods in System Design

A Compositional Modelling and Analysis Framework for Stochastic Hybrid Systems

Ernst Moritz Hahn · Arnd Hartmanns · Holger Hermanns · Joost-Pieter Katoen

model f

# The Modest Toolset



mime

Modest

PRISM Guarded Commands

UPPAAL .xml

Networks of Stochastic Hybrid Automata

SHA

STA

PTA

MDPs

prohver

modes

mcpta

mctau

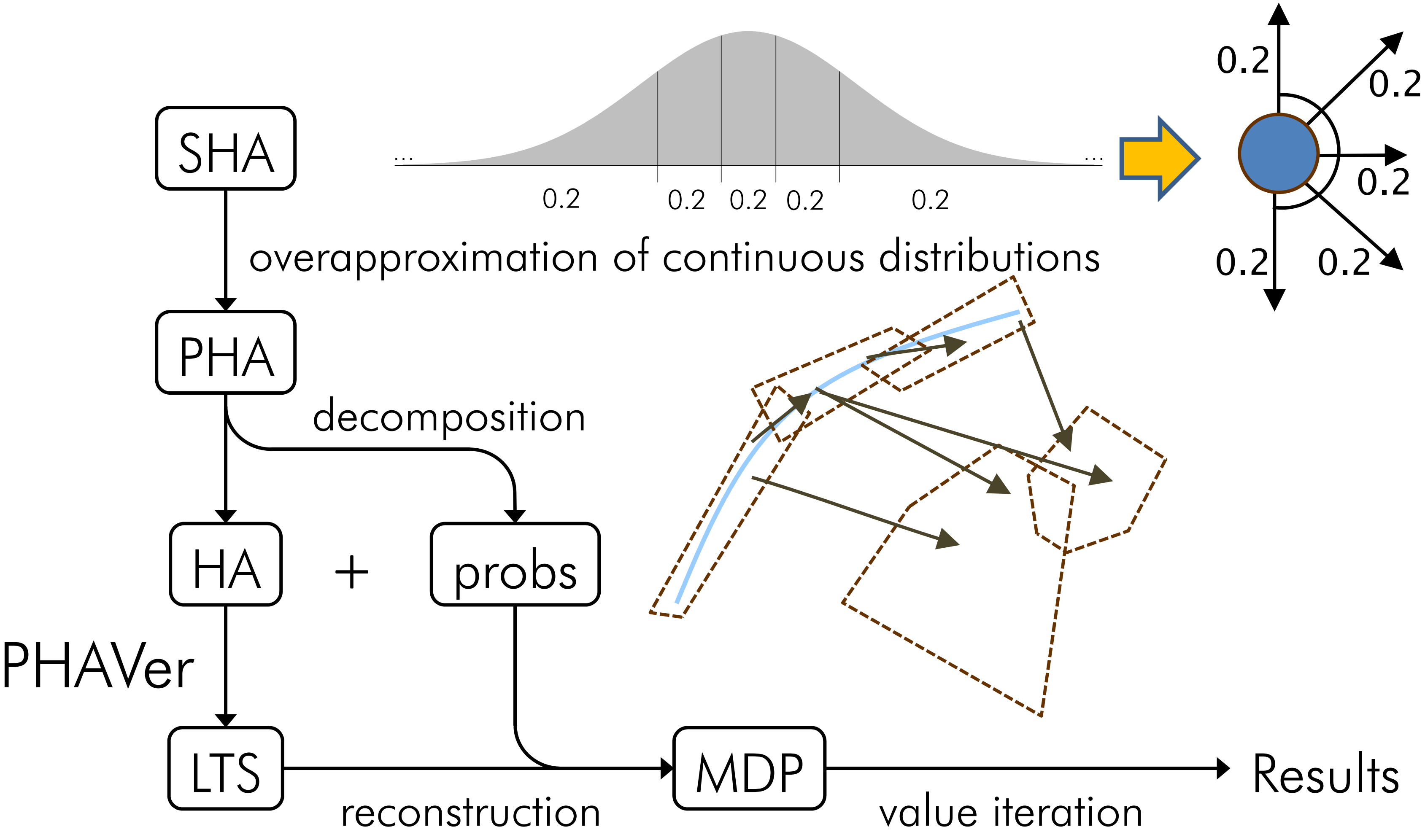
modified PHAVer

PRISM ≥ 4.0

UPPAAL ≥ 4.1

# prohver

## Safety verification process for SHA in prohver



SHA

overapproximation of continuous distributions

PHA

decomposition

HA + probs

PHAVer

LTS

reconstruction

MDP

value iteration

Results

# Case Study - ETCS level 3

## SHA model:

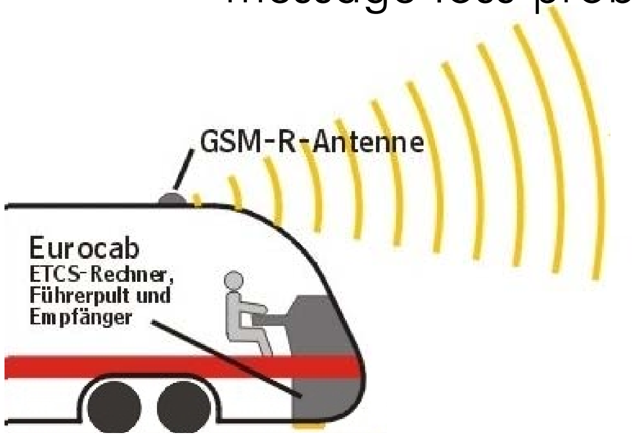
- two trains – leader and follower – and Comm+RBC

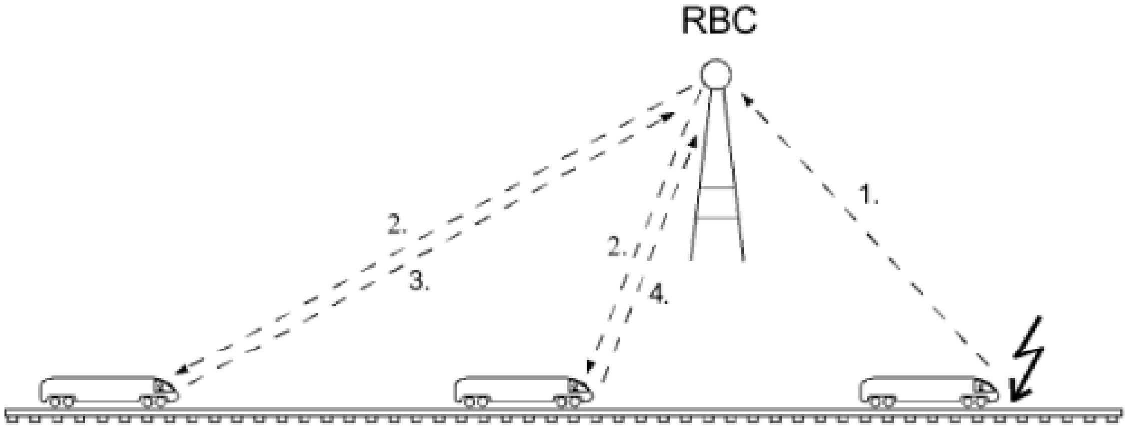
## Continuous aspects

- acceleration, deceleration, speed
- acceleration of leader nondeterministic (within train limits)

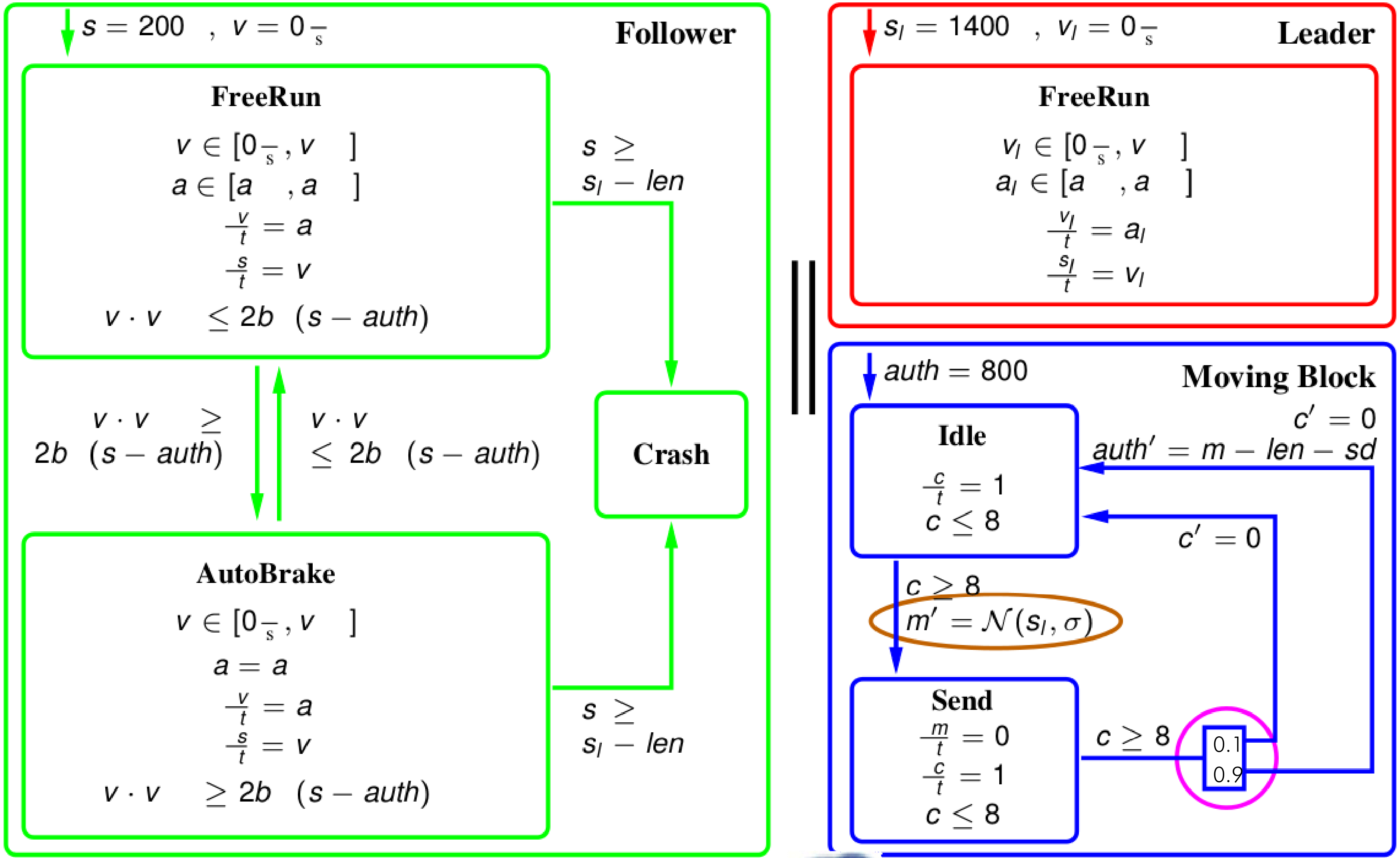
## Stochastic aspects

- position measurements scattered with normal distribution
- message loss probability during communication





# Case Study - ETCS level 3



Follower

$s = 200 \quad , \; v = 0\frac{}{\text{s}}$

**FreeRun**

$v \in [0\frac{}{\text{s}}, v \quad ]$

$a \in [a \quad , a \quad ]$

$\frac{v}{t} = a$

$\frac{s}{t} = v$

$v \cdot v \quad \leq 2b \;\; (s - auth)$

$v \cdot v \quad \geq$ $2b \;\; (s - auth)$

$v \cdot v$ $\leq \; 2b \;\; (s - auth)$

**AutoBrake**

$v \in [0\frac{}{\text{s}}, v \quad ]$

$a = a$

$\frac{v}{t} = a$

$\frac{s}{t} = v$

$v \cdot v \quad \geq 2b \;\; (s - auth)$

$s \geq$ $s_l - len$

**Crash**

$s \geq$ $s_l - len$

$\|$

Leader

$s_l = 1400 \quad , \; v_l = 0\frac{}{\text{s}}$

**FreeRun**

$v_l \in [0\frac{}{\text{s}}, v \quad ]$

$a_l \in [a \quad , a \quad ]$

$\frac{v_l}{t} = a_l$

$\frac{s_l}{t} = v_l$

Moving Block

$auth = 800$

**Idle**

$\frac{c}{t} = 1$

$c \leq 8$

$c' = 0$

$auth' = m - len - sd$

$c' = 0$

$c \geq 8$

$m' = \mathcal{N}(s_l, \sigma)$

**Send**

$\frac{m}{t} = 0$

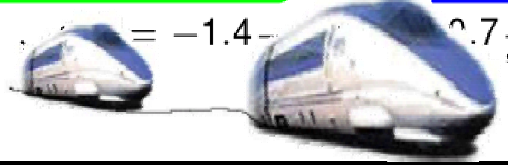
$\frac{c}{t} = 1$

$c \leq 8$

$c \geq 8$

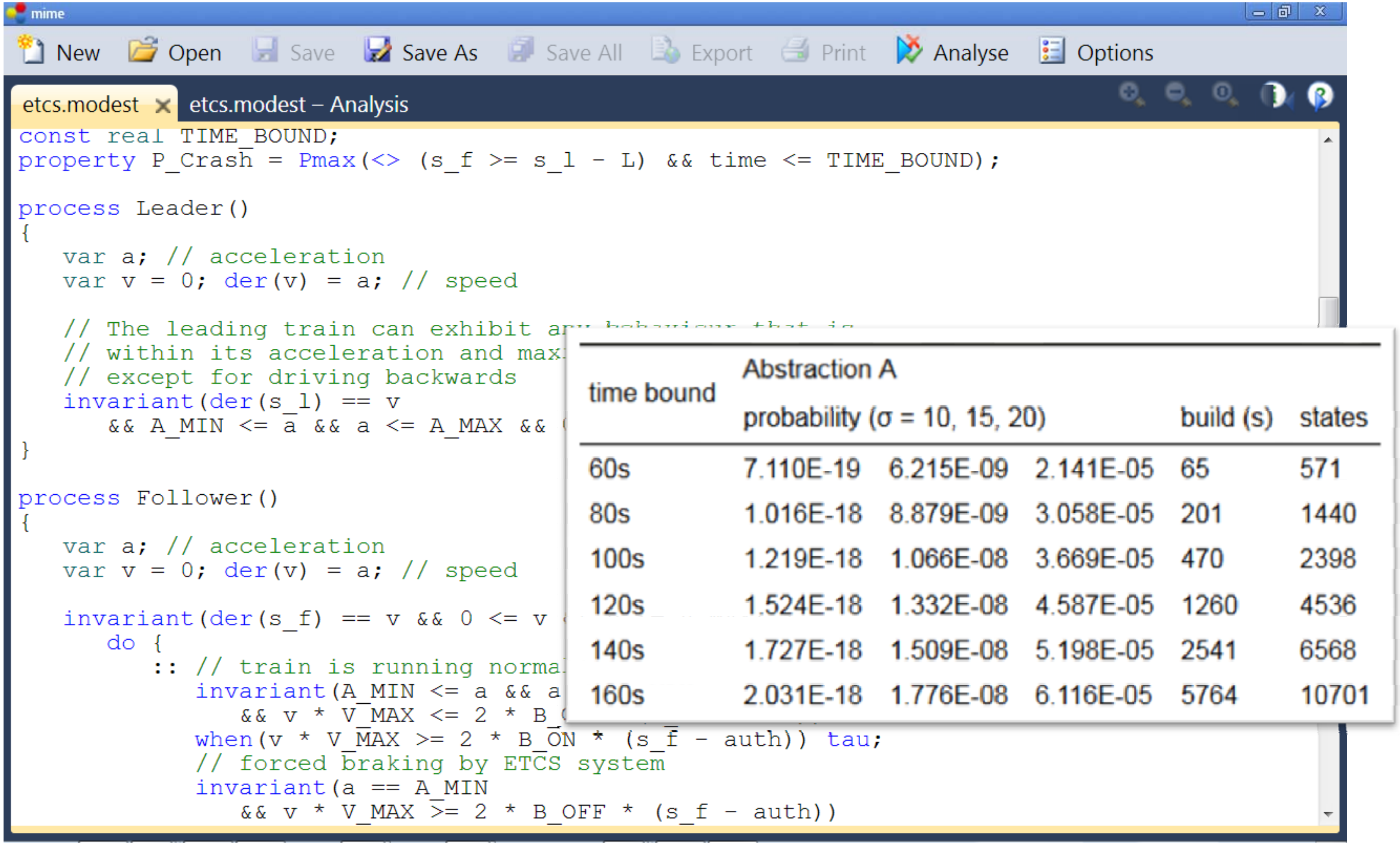
0.1

0.9



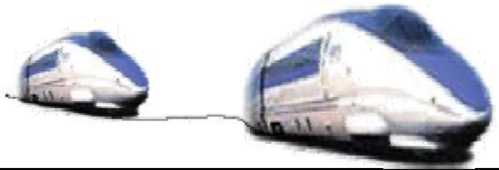
$v \quad = 83.4\frac{}{\text{s}}, \;\; len = 200 \quad , \;\; sd = 400 \quad . \quad = -1.4 \quad .7\frac{}{\text{s}^2}, \;\; b \quad = -0.7\frac{}{\text{s}^2}, \;\; b \quad = -0.3\frac{}{\text{s}^2}$

# Case Study - ETCS level 3

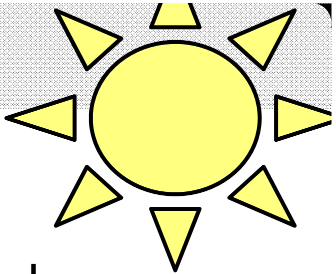


```
const real TIME_BOUND;
property P_Crash = Pmax(<> (s_f >= s_l - L) && time <= TIME_BOUND);

process Leader()
{
   var a; // acceleration
   var v = 0; der(v) = a; // speed

   // The leading train can exhibit any behaviour that is
   // within its acceleration and max
   // except for driving backwards
   invariant(der(s_l) == v
      && A_MIN <= a && a <= A_MAX && (
}

process Follower()
{
   var a; // acceleration
   var v = 0; der(v) = a; // speed

   invariant(der(s_f) == v && 0 <= v (
      do {
         :: // train is running normal
            invariant(A_MIN <= a && a
               && v * V_MAX <= 2 * B_(
            when(v * V_MAX >= 2 * B_ON * (s_f - auth)) tau;
            // forced braking by ETCS system
            invariant(a == A_MIN
               && v * V_MAX >= 2 * B_OFF * (s_f - auth))
```

| time bound | Abstraction A | | | | |
|---|---|---|---|---|---|
| | probability (σ = 10, 15, 20) | | | build (s) | states |
| 60s | 7.110E-19 | 6.215E-09 | 2.141E-05 | 65 | 571 |
| 80s | 1.016E-18 | 8.879E-09 | 3.058E-05 | 201 | 1440 |
| 100s | 1.219E-18 | 1.066E-08 | 3.669E-05 | 470 | 2398 |
| 120s | 1.524E-18 | 1.332E-08 | 4.587E-05 | 1260 | 4536 |
| 140s | 1.727E-18 | 1.509E-08 | 5.198E-05 | 2541 | 6568 |
| 160s | 2.031E-18 | 1.776E-08 | 6.116E-05 | 5764 | 10701 |



# Case Study - Power Grid Control Strategies



All over Germany,
masses of photovoltaic microgenerators are rolled out:

2009: 10 GW    2011: 25 GW    2020: ?? GW

**Current state of control:**

EN 50438:2007, in force since 2007:

Switch off when frequency $> 50.2\,\text{Hz}$

**"on-off" controller**

VDE-AR-N 4105, required today:

Output linear function of frequency in $[50.2, 51.5]\,\text{Hz}$

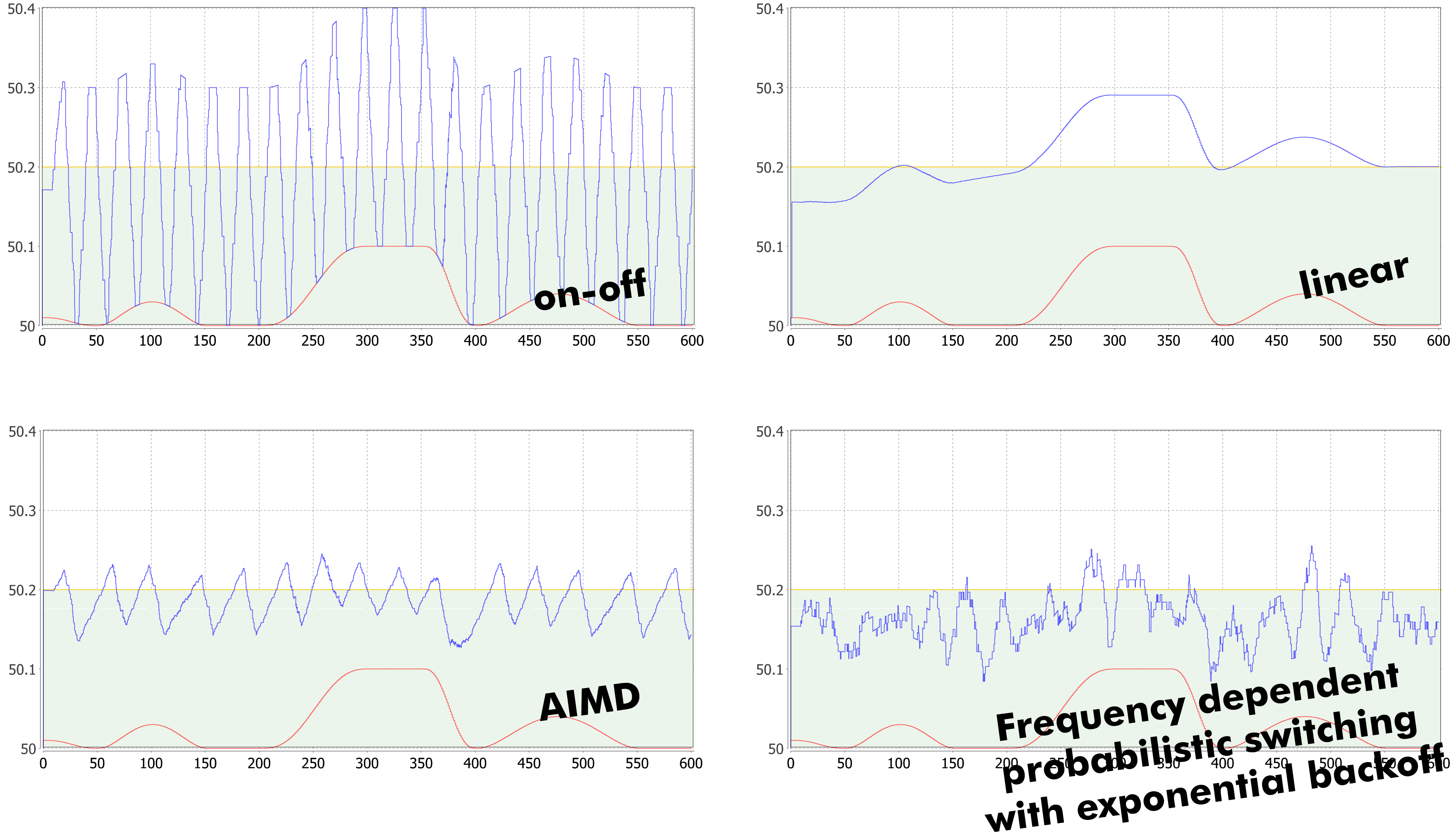
Emergency switchoff above $51.5\,\text{Hz}$

Switch on again when $< 50.05\,\text{Hz}$ for 1 minute

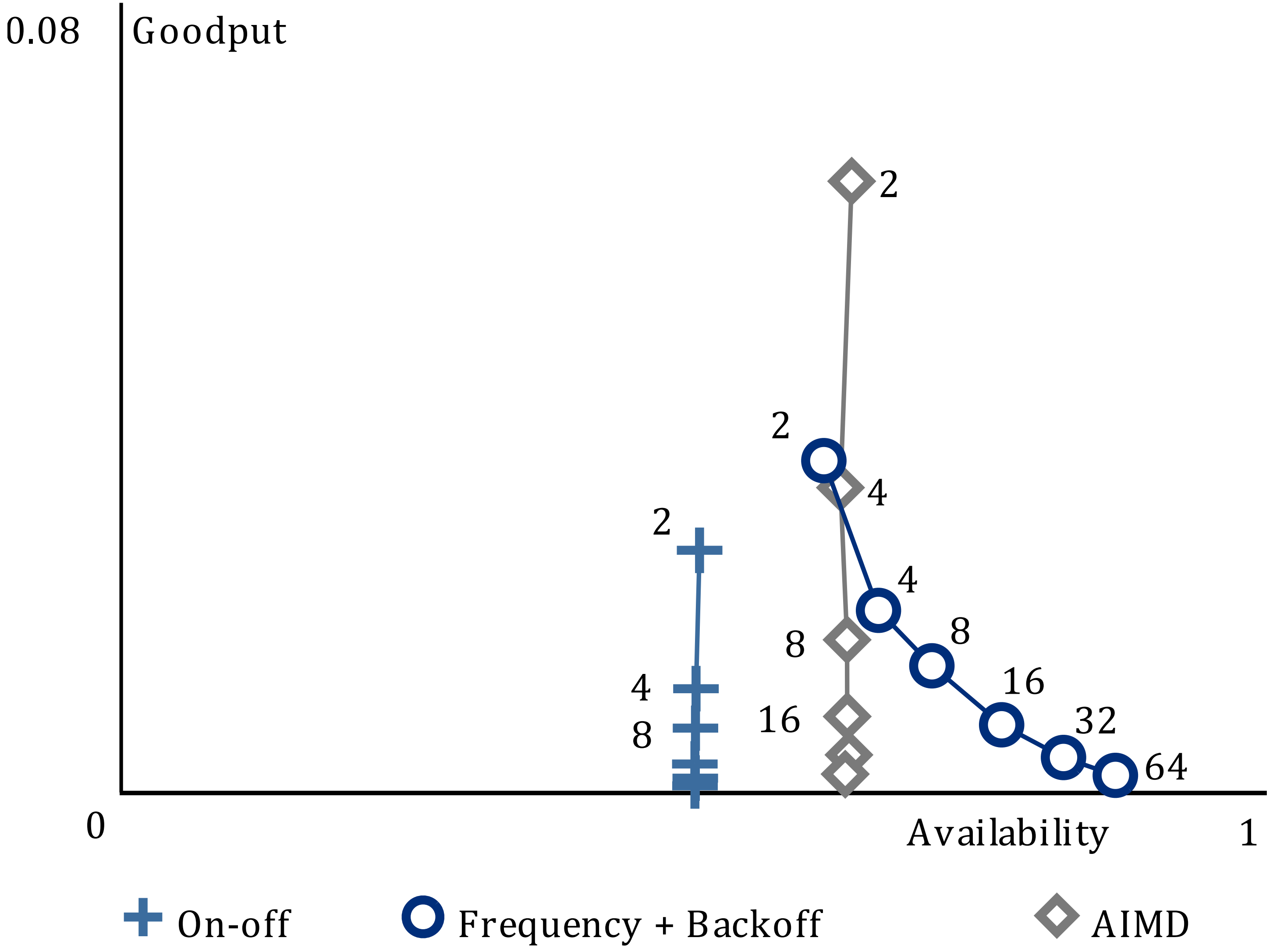
**"linear" controller**

# Stability of Grids and Controllers

## Simulation of synthetic background load scenarios

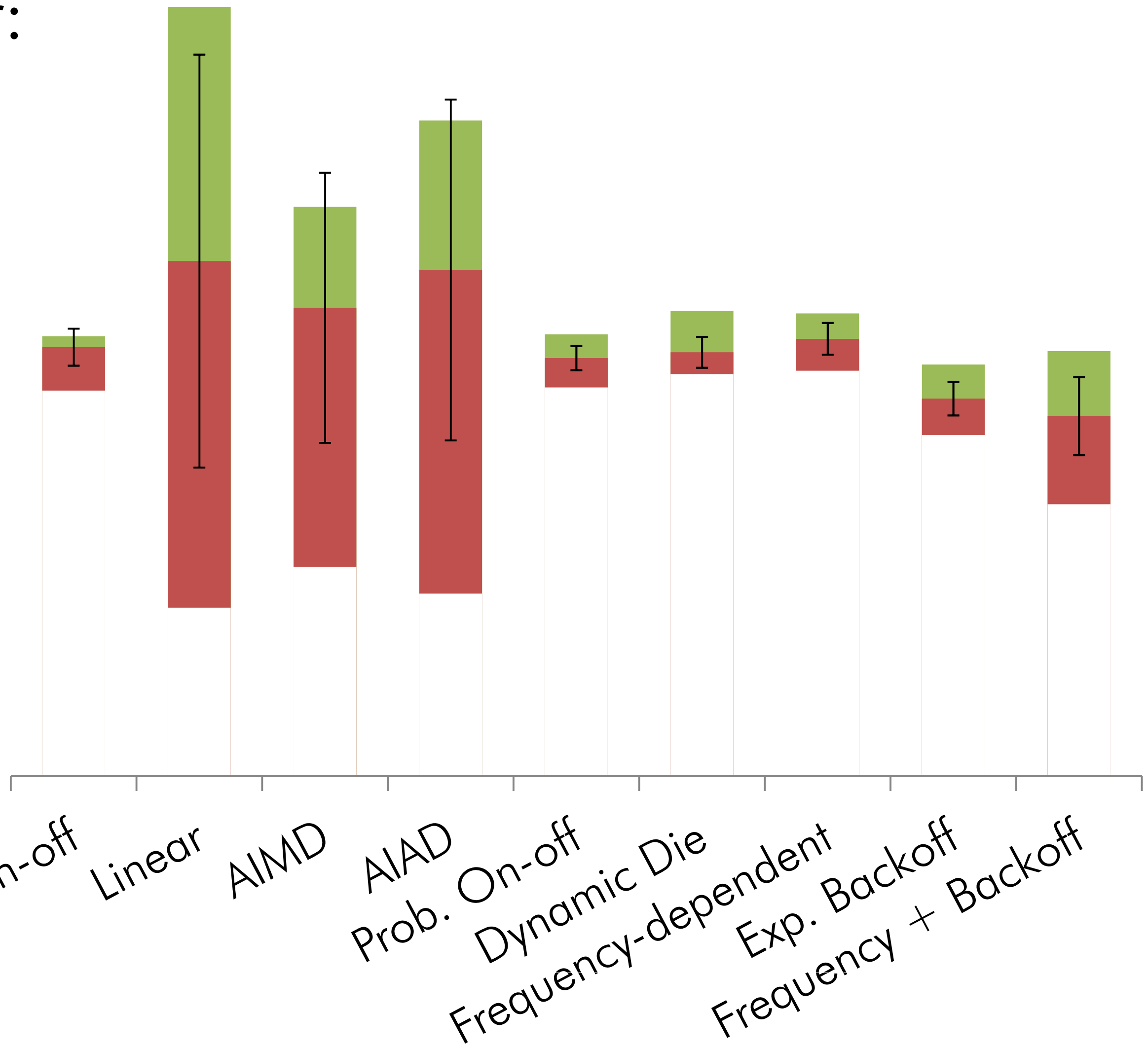


# Availability vs. Goodput

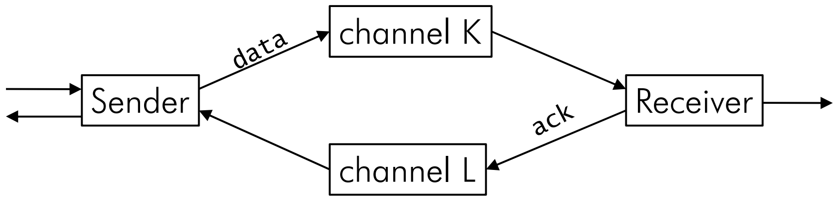


# Fairness of Controllers

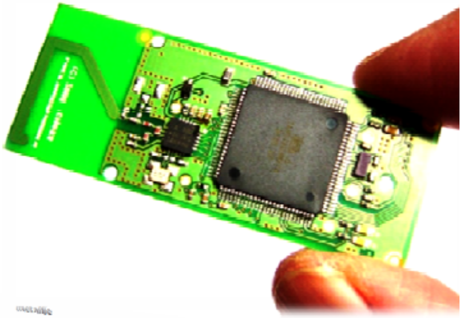
Max/min/average output per generator:



# Modest Applications

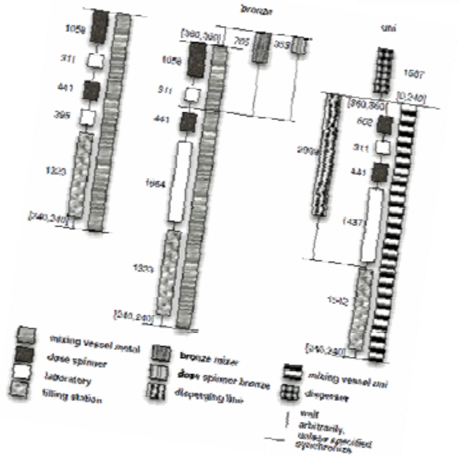


Communication protocols



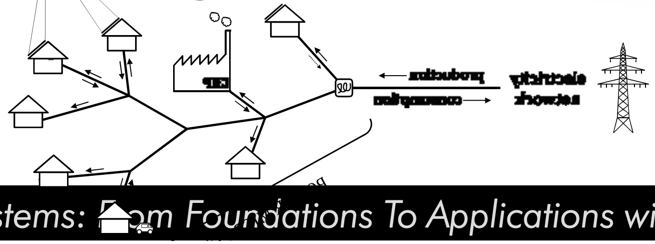
Wireless sensor networks

Dependability evaluation



Industrial production scheduling

Renewable electric power generation



# Modest - Summary

Modest and SHA
– language and model
for quantitative systems
with quantitative requirements

```
{= x = Uni(0, 3) =}
```

$E_{max}$ [time to finish]

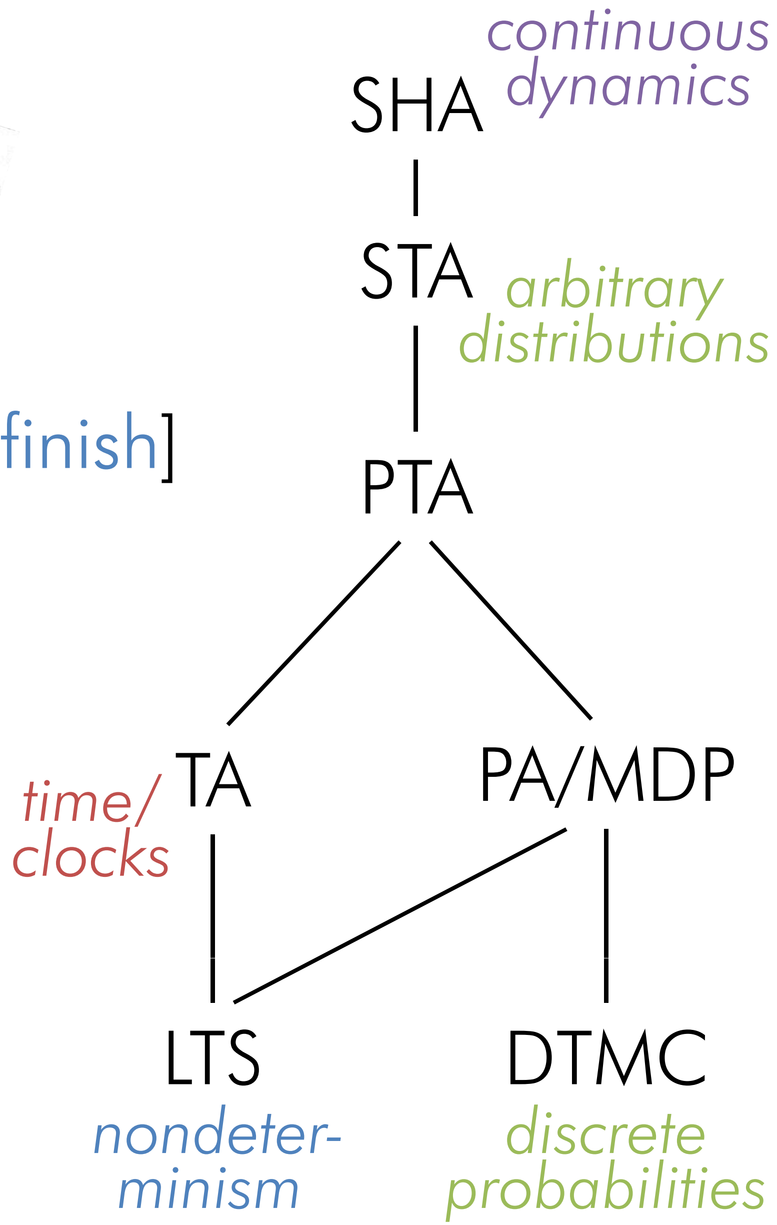
```
var v, a;
invariant(der(v) == a) …
```

```
invariant(c <= TD_MAX)
```

```
par {
:: Sender()
:: P()
}
```

```
snd palt {
  :99: rcv
  : 1: tau }
```

⇨ *single-formalism, multiple-solution approach*



# The Modest Toolset - Summary

modelling language: Modest
+ PRISM guarded commands
+ UPPAAL xml

| | |
|---|---|
| **prohver** | for SHA - using Phaver |
| **mcpta** | for PTA/MDP - using PRISM |
| **mctau** | for TA - using UPPAAL |
| **modes** | for simulation despite nondeterminsm |

Demo at demo session on Friday!

Installation assistance anytime!

www.modestchecker.net